Scheduling of Mask Shop E-Beam Writers

Yi-Feng Hung

***Abstract*—Reducing wafer fabrication cycle time and providing on-time wafer deliveries are among the top priorities of semiconductor companies. Mask manufacturing is essential to the overall wafer fabrication process since on-time delivery of masks significantly affects wafer fabrication cycle times. Moreover, delivering wafers on time means deliveries of masks must be on time as well. This research studies the scheduling problem of the bottleneck machine—the Electrical Beam (E-beam) Writer—of a mask shop. The criterion of minimum total tardiness is used as our performance measure to schedule this bottleneck operation. Using a predetermined Earliest-Due-Date (EDD) dispatch policy set by management, this study first addresses the problem of scheduling batches of a single mask size on a single machine. The approach is extended to the problem of scheduling batches of two mask sizes on a single machine; finally, a heuristic for a multiple-machine problem is developed. For the problem of a single machine under EDD dispatching policy, the problem can be formulated as a Dynamic Program (DP). Thus, it can be solved for an optimal solution in polynomial time. For the multiple machines problem, we heuristically allocate the masks to each machine. Each machine with allocated masks can then be solved by the DP formulation designed for the single machine problem. Based on the computational experiments in this study, the proposed DP approach reduces total tardiness by an average of 55% from the method currently in use at a major IC manufacturing foundry. Furthermore, in the case that due dates are set realistically, the DP approach reduces the tardiness about 95% from the shop's current method and about 88% from a simple full-batch method of scheduling.**

***Index Terms*—Batch scheduling, dynamic programming, mash shop scheduling.**

I. Introduction

WAFER fabrication factories insist that mask shops deliver masks on time since delayed mask deliveries mean delayed introduction of products to markets or delayed deliveries to customers. The semiconductor product prices drop rapidly; thus, reducing the overall cycle time from design to sale can have a significant impact on company profitability. Also, on-time delivery is a performance criterion many companies strive to improve. Thus, mask manufacturing plays a prominent role in wafer fabrication. Owing to its high capital investment, the Electrical Beam (E-beam) Writer, which defines mask geometric patterns, is usually the bottleneck machine in mask shops.

Meeting due dates for the masks with confirmed deliveries is a mask shop's highest priority. The case under study is an internal mask shop in a major foundry semiconductor firm. The dispatch rule for all machines in the mask shop set by management is the Earliest-Due-Date (EDD) rule. According to this rule, a mask having a later due date cannot be processed before a mask with an earlier due date. The bottleneck machine (the E-beam writer) is the first operation in the mask manufacturing process. This study concentrates only on this E-beam machine, since the performance of the entire mask shop is primarily determined by this bottleneck machine. In focusing on this subsystem, which consists of only E-beam writers, the due date of each mask in this subsystem is computed by subtracting the average flow time of the remaining operations from the actual due date of the mask. This average flow time is relatively stable owing to the fact that the remaining operations are performed by nonbottleneck machines. After discussing the performance measure with the management, we attempted to minimize the total tardiness of all masks with confirmed deliveries because meeting due dates of all confirmed masks were considered to be equally important. There are other due-date related criteria, which were not used. The measure of number of tardy jobs was not used because extreme prolonging of the delivery of a few jobs in exchange for on-time delivery of other masks was not considered acceptable. Total lateness criterion was not used because the degree to which some jobs are early cannot make up for delays in tardy jobs.

The E-beam operation involves grouping several masks into one batch, then sealing the chamber into which the batch is placed. After evacuating air from the chamber, the E-beam writer then starts writing the geometric patterns on the masks one-by-one. The processing time for writing each mask can be estimated from the historical database and is assumed here to be a known constant parameter. After patterns have been written on all masks in the batch, the chamber is vented to atmosphere, and an operator removes the completed batch. Since the chamber of E-beam writer cannot be opened until all masks in a batch are completed, the completion times for all of the masks in a given batch are the same. We define setup time of the batch to include operator load, pump to vacuum, vent to atmosphere and operator unload times. Thus, the total processing time for a batch is the setup time plus the sum of the processing times for all masks in the batch.

There is a capacity limitation for the chamber; the number of masks in a batch cannot exceed this limit. Furthermore, there are currently two mask sizes—5-in and 6-in—which by company policy cannot be mixed in the same batch. That is, all masks in one batch must be of the same size. The additional setup involved in switching between 5- and 6-in masks can be neglected and the batch setup times are assumed to be the same for both mask sizes.

Using the minimum total tardiness performance measure, the tradeoff in E-beam processing involves the dynamic sizing of batches. If we make a certain batch larger, we complete more masks per unit of time, which benefits masks processed afterwards, since their start times can be earlier and, therefore,

Manuscript received November 14, 1996; revised September 22, 1997. This work was supported in part by the National Science Council and the Taiwan Semiconductor Manufacturing Company, Taiwan.

The author is with the Department of Industrial Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C. (e-mail: yifeng@ie.nthu.edu.tw).

Publisher Item Identifier S 0894-6507(98)00331-5.

their tardiness is reduced. On the other hand, the tardiness of masks in the current batch will be increased, because they will be completed later. A tradeoff argument can be made for making a certain batch smaller. That will reduce its tardiness; however, masks processed afterwards will be more tardy. To resolve this tradeoff, this study focuses on how to achieve optimal batch decision that minimizes total tardiness.

When this study was initiated, the mask shop had only one E-beam writer, and was expecting to receive another later on. Therefore, this study began as a single-machine problem, and was then extended to a multiple-machine problem.

Koulamas [12] reviewed scheduling methods for the total tardiness problem. That the complexity of a single-machine deterministic total-tardiness problem is NP-hard in the ordinary sense was proved by Du and Leung [5]. Our problem, if not sorted using EDD, is more complex than the conventional single-machine total-tardiness problem and, thus, is also an NP-hard problem. However, the EDD dispatch rule set by management narrows the solution domain and simplifies our problem. As is shown below, Dynamic Programming (DP) formulations with complexities of $O(n^2)$ and $O(n^3)$ can be used for a one-mask-size problem and a two-mask-size problem, respectively. Consequently, the single-machine problem under study is not an NP-hard problem.

Overall semiconductor manufacturing involves several types of batch operations, the processing times for which are defined differently largely due to equipment characteristics. Besides the problem discussed in this paper, there are at least two other types of batch problems in semiconductor manufacturing. One is that processing time is determined by the product type in a batch and is independent of how many jobs are in the batch. The furnace tubes used for the deposition operation in wafer fabrication is a typical example. For the work on scheduling this kind of batch problem, see [6], [7], [10], and [16]. The other one is that the processing time of a batch is equal to the processing time of the longest job in the batch, which is exemplified by the problem of scheduling burn-in ovens for back-end test operations. A series of works on this problem is presented in [2], [3], [13], and [14]. To the best of the author's knowledge, no work has previously been done on the batch scheduling problem for the E-beam writer presented in this paper.

II. Dynamic Programming for a Single Mask Size

The notation needed for this study is as follows:

- u Batch setup time.
- b Maximum batch size; the largest number of masks that can be processed in one batch.
- s Setup (batch) number.
- J Total number of masks to be scheduled.
- j Mask (job) number after sorting according to the Earliest-Due-Date (EDD) rule; $j = 1, 2, \ldots, J$.
- p_j Processing time of the jth mask.
- d_j Due date of the jth mask.
- $s(j)$ Batch number of the jth mask in a certain schedule.
- C_s Completion time for the sth batch.
- t_j Tardiness of the jth mask in a certain schedule; $t_j = \max(0, C_{s(j)} - d_j)$.
- T Total tardiness in a certain schedule; $T = \sum_{j=1}^{J} t_j = \sum_{j=1}^{J} \max(0, C_{s(j)} - d_j)$.
- $\underline{s}$ Minimum number of batches.
- $\overline{s}$ Maximum number of batches.

In the following two subsections, an invalid problem formulation and a valid formulation are presented. The invalid formulation attempts to demonstrate that the conventional Wagner-Whitin formulation for the Dynamic Lot Sizing Problem [15] cannot be applied here.

A. Invalid Problem Formulation

After the mask sequence has been sorted according to the EDD rule, the batch decision seems to be similar to the Wagner-Whitin dynamic lot sizing problem [15].

In addition to the notation defined previously, the state variable for the DP formulation is

- j current mask number.

The parameters that can be obtained from previous recursive computations of the DP formulation are

- C_j^* completion time of the first j masks using the optimal batch decision determined previously to process the first j masks.

The decision variable to be used in the DP formulation is

- k size of the last batch.

The recursive function notation in the DP formulation is

- $T(j)$ minimum total tardiness of processing the first j masks.

The recursive relationship can then be defined as

$$T(j) = \min_{k=1,2,\ldots,b'_j} \left\{ T(j-k) + \sum_{l=j-k+1}^{j} \max\left(0, C^*_{j-k} + u + \sum_{i=j-k+1}^{j} p_i - d_l\right)\right\}$$

where

$$b'_j = \min\{b, j\}.$$

In the recursive relationship, the expression $C^*_{j-k} + u + \sum_{l=j-k+1}^{j} p_i$ is the completion time if we group k masks in the last batch and use the optimal policy to process the first $j-k$ masks. Therefore, the expression $\sum_{l=j-k+1}^{j} \max(0, C^*_{j-k} + u + \sum_{i=j-k+1}^{j} p_i - d_l)$ is the tardiness of the last batch, whose size is k. The term $T(j-k)$ is the optimal tardiness value of the previous $(j-k)$ masks. The parameter b'_j is the maximum possible value for the size of last batch, which is the minimum of the batch size of the E-beam chamber (b) and the number of mask scheduled thus far (j).

Finding the solution requires computing $T(j)$ from $j = 1$ to J.

It is clear that this formulation is similar to the Wagner-Whitin model. But, it is unfortunate that applying this formulation to our problem violates the principle of optimality for DP given by Hillier and Liberman [8]:

"*Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages. Therefore, the optimal immediate decision depends on only the current state and not on how you got there.*"

By examining the previous formulation, given the current state variable j (the number of masks scheduled thus far), the optimal policy for the remaining masks does depend on how we got to j. The total completion time for the first j masks will affect batch decisions concerning the remaining masks, but this total completion time information cannot be provided by the state variable used. Being more specific, we may have an optimal batch decision that minimize the total tardiness for the current j masks, but it may result in a larger total completion time. However, this larger completion time will make the masks scheduled afterwards more tardy. Furthermore, by applying the principle pointed out by Dreyfus and Law [4] to this formulation, we can find that it fails to satisfy the "consultant question" criteria; that is, the DP state variable (argument) has not been properly chosen. To clarify, suppose that we are the consultants referred to Dreyfus and Law's principle, and are trying to take over the problem. Unfortunately, merely knowing the number of masks scheduled thus far and, of course, their optimal total tardiness value, we cannot reach the optimal decision for the remaining masks because examining the state variable does not tell us the completion times for the previously scheduled masks. Thus, the minimum information we require to take over the remaining problem is not merely the number of masks, we also need information (state variable) concerning the number of setups already used. Therefore, this problem cannot be formulated as a conventional Wagner-Whitin model, that has only one state variable, and the above DP is invalid and requires modification.

B. Valid Problem Formulation

Here, another state variable (s; the number of setups) must be added for a valid formulation of the problem. Therefore, the state variables required are

s number of setups (batches) used.
j current mask number.

The parameters needed in addition to those mentioned in Section II-A are

$C_{j,s}$ total completion time for processing the first j masks using s setups.

The decision variable is k (number of masks in the last batch), as in Section II-A.

The modified recursive function notation is

$T(j,s)$ minimum total tardiness for processing the first j masks using s setups.

The recursive function for the valid DP formulation is

$$T(j,s) = \min_{k=1,2,\ldots,b'_j} \left\{ T(j-k, s-1) + \sum_{l=j-k+1}^{j} \max(0, C_{j,s} - d_l) \right\},$$

where

$$C_{j,s} = s \cdot u + \sum_{i=1}^{j} p_i;$$
$$b'_j = \min\{b, j\}.$$

Note that $C_{j,s}$ is independent of how masks are grouped into batches. Given j masks to be processed in s setups, the choice we face is the number of masks in the very last batch. The optimal policy is determined by selecting the minimum total tardiness among all candidates. The decision variable k in the formulation is the number of masks in the last batch, which can range from 1 to either the maximum batch size (b) or the current number of masks (j), whichever is smaller. Given j masks and s setups, if we group k masks in the last batch, the tardiness of the last batch and the optimal tardiness value for processing first $j-k$ masks with $s-1$ setups must be summed to determine the total tardiness. The term $T(j-k, s-1)$ in the recursive function is the optimal value of the previous $(j-k)$ masks in $(s-1)$ batches, and the expression $\sum_{l=j-k}^{j} \max(0, C_{j,s} - d_l)$ is used to compute the tardiness of the last batch. Thus, the optimal policy involves selecting the optimal last batch size k to achieve minimum total tardiness; $k^*(j,s)$ is defined to record the optimal path:

$k^*(j,s)$ optimal last batch size for processing the first j masks using s setups.

When finding the optimal schedule of the first j masks, the upper bound and lower bound functions must be defined for the feasible number of batches:

$$\underline{s} = \left\lceil \frac{j}{b} \right\rceil \quad \text{and} \quad \overline{s} = j,$$

where $\lceil \frac{j}{b} \rceil$ is the smallest integer greater than $\frac{j}{b}$; thus, it is the minimum number of batches required to process j masks. The maximum possible number of batches is equal to the number of masks—j.

In addition, the boundary conditions are

$$T(j,s) = \infty, \quad \text{if } s < \underline{s} \quad \text{or} \quad s > \overline{s},$$

and

$$T(j,s) = 0, \quad \text{if } j = 0.$$

If the number of setups s is smaller than the minimum number of batches required or greater than the number of masks, it cannot be on the optimal path of the problem; therefore, we set its tardiness equal to infinity to speed up computation. Additionally, if there are no masks requiring processing, the tardiness is zero.

After defining the recursive function and boundary conditions, the optimal total tardiness can be obtained by

$$T^*(J) = \min_{s=\underline{s},\ldots,\overline{s}} \{T(J,s)\}$$

which chooses the optimal number of batches (setups) to minimize the total tardiness.

C. Other Approaches to Solve the Tardiness Problem

Before this study, the mask shop used a heuristic method we shall call the *Dynamic Fixed Batch approach* (DFB), in which an estimated batch size is determined at scheduling time by computing

$$f = \frac{Q}{\frac{D \cdot m - P}{u}}$$

where Q is the total number of masks to be scheduled, D is the maximum due date among all masks, m is the number of machines, P is the total processing time of all masks, and u is the setup time. In the formula, $D \cdot m - P$ is the estimated total slack time for setup, if the last mask (with latest due date) is to be completed on time; thus, the denominator of the formula is the estimated number of setups available. If the estimated batch size (f) is negative or greater than the maximum batch size (b), the fixed batch size is set at the maximum batch size. Otherwise, the fixed batch size is set at the nearest integer to f. Then, grouping the masks according the fixed size of f is used.

Another method we shall call the *Full Batch Approach* (FB) was also used in this study to compare with the above two approaches. In this approach, the batches are always fully loaded with masks, except the last batch may not have enough masks to be full. (The author is indebted to the referee for suggesting this approach.)

D. A Numerical Example for Single Mask Size on Single Machine

A simple numerical example with a single mask size on a single machine is used to demonstrate the methods presented above. We let $u = 5$, $b = 3$, $J = 5$, $p_1 = 3$, $d_1 = 6$, $p_2 = 4$, $d_2 = 13$, $p_3 = 5$, $d_3 = 27$, $p_4 = 6$, $d_4 = 30$, $p_5 = 7$, $d_5 = 45$.

1) Dynamic Fixed Batch Approach: The estimated batch size is

$$f = \frac{Q}{\frac{D \cdot m - P}{u}} = \frac{5}{\frac{45 - (3+4+5+6+7)}{4}} = 1.25$$

and its nearest integer is 1. Therefore, the fixed batch size is 1. Thus, the completion times are $C_1 = u + p_1 = 8$, $C_2 = u + p_1 + u + p_2 = 17$, $C_3 = u + p_1 + u + p_2 + u\ p_3 = 27$, $C_4 = u + p_1 + u + p_2 + u + p_3 + u + p_4 = 38$, and $C_5 = u + p_1 + u + p_2 + u + p_3 + u + p_4 + u + p_5 = 50$. Then, the total tardiness is $T_1 + T_2 + T_3 + T_4 + T_5 = \max(0, C_1 - d_1) + \max(0, C_2 - d_2) + \max(0, C_3 - d_3) + \max(0, C_4 - d_4) + \max(0, C_5 - d_5) = \max(0, 8-6) + \max(0, 17-13) + \max(0, 27-27) + \max(0, 38-30) + \max(0, 50-45) = 19$.

2) Full Batch Approach: In this method, the first batch contains mask 1 to mask 3 and their completion time is $C_1 = u + p_1 + p_2 + p_3 = 5 + 3 + 4 + 5 = 17$; thus, their tardiness $T_1 + T_2 + T_3 = \max(0, C_1 - d_1) + \max(0, C_1 - d_2) + \max(0, C_1 - d_3) = \max(0, 17-6) + \max(0, 17-13) + \max(0, 17-27) = 15$. The second batch consists of mask 4 and 5 and their completion time $C_2 = u + p_1 + p_2 + p_3 + u + p_4 + p_5 = 5 + 3 + 4 + 5 + 5 + 6 + 7 = 35$; thus, their tardiness $T_4 + T_5 = \max(0, C_2 - d_4) + \max(0, C_2 - d_5) = \max(0, 35-30) + \max(0, 35-45) = 5$. Therefore, the total tardiness of the schedule obtained by this approach is 20 20 $(=15 + 5)$.

3) Dynamic Programming Approach: In this approach, we shall iterate over the possible number of masks (j) from 1 to 5, and, in each value of j, we shall iterate over all the possible number of setups.

Initially, we set all $T(j, s) = \infty$ for $j = 1$ to 5 and for $s = 1$ to 5.

When $j = 1$: the only possible number of setup is 1. We can compute $C_{1,1} = u + p_1 = 5 + 3 = 8$, then the optimal tardiness $T(1,1) = \max(0, C_{1,1} - d_1) = 2$ and the optimal last batch size $k^*(1,1) = 1$.

When $j = 2$: the possible number of setup is from 1 to 2.

In the case of $j = 2$ and $s = 1$: we can compute $C_{2,1} = u + p_1 + p_2 = 5 + 3 + 4 = 12$, $T(2,1) = \max(0, C_{2,1} - d_1) + \max(0, C_{2,1} - d_2) = 6$ and $k^*(2,1) = 2$.

In the case of $j = 2$ and $s = 2$: we can compute $C_{2,2} = u + u + p_1 + p_2 = 5 + 5 + 3 + 4 = 17$, then

$$\begin{aligned} T(2,2) &= \min\{T(1, 1+\max(0, C_{2,2} - d_2), \\ &\qquad T(0,1) + \max(0, C_{2,2} - d_1) + \max(0, C_{2,2} - d_2)\} \\ &= \min\{2 + 4, \infty + 11 + 4\} = 6. \end{aligned}$$

Thus, $k^*(2,2) = 1$.

When $j = 3$: the possible number of setup(s) is from 1 to 3:

In the case of $j = 3$ and $s = 1$: we can compute $C_{3,1} = u + p_1 + p_2 + p_3 = 5 + 3 + 4 + 5 = 17$, $T(3,1) = \max(0, C_{3,1} - d_1) + \max(0, C_{3,1} - d_2) + \max(0, C_{3,1} - d_3) = 11 + 4 + 0 = 15$ and $k^*(3,1) = 3$.

In the case of $j = 3$ and $s = 2$: we can compute $C_{3,2} = u + u + p_1 + p_2 + p_3 = 5 + 5 + 3 + 4 + 5 = 22$, then

$$\begin{aligned} T(3,2) &= \min\{T(2,1) + \max(0, C_{3,2} - d_3), \\ &\qquad T(1,1) + \max(0, C_{3,2} - d_2) + \max(0, C_{3,2} - d_3), \\ &\qquad T(0,1) + \max(0, C_{3,2} - d_1) + \max(0, C_{3,2} - d_2) \\ &\qquad + \max(0, C_{3,2} - d_3)\} \\ &= \min\{6 + 0, 2 + 9 + 0, \infty + 16 + 9 + 0\} = 6. \end{aligned}$$

Thus, $k^*(3,2) = 1$.

In the case of $j = 3$ and $s = 3$: we can compute $C_{3,3} = u + u + u + p_1 + p_2 + p_3 = 5 + 5 + 5 + 3 + 4 + 5 = 27$, then

$$\begin{aligned} T(3,3) &= \min\{T(2,2) + \max(0, C_{3,3} - d_3), \\ &\qquad T(1,2) + \max(0, C_{3,3} - d_2) + \max(0, C_{3,3} - d_3), \\ &\qquad T(0,2) + \max(0, C_{3,3} - d_1) + \max(0, C_{3,3} - d_2) \\ &\qquad + \max(0, C_{3,3} - d_3)\} \\ &= \min\{6 + 0, \infty + 14 + 0,\ \infty + 21 + 14 + 0\} = 6. \end{aligned}$$

Thus, $k^*(3,3) = 1$.

When $j = 4$: the possible number of batch (s) is from 2 to 4:

In the case of $j = 4$ and $s = 2$: we can compute $C_{4,2} = u + u + p_1 + p_2 + p_3 + p_4 = 5 + 5 + 3 + 4 + 5 + 6 = 28$, then

$$\begin{aligned} T(4,2) &= \min\{T(3,1) + \max(0, C_{4,2} - d_4), \\ &\qquad T(2,1) + \max(0, C_{4,2} - d_3) + \max(0, C_{4,2} - d_4), \\ &\qquad T(1,1) + \max(0, C_{4,2} - d_2) + \max(0, C_{4,2} - d_3) \\ &\qquad + \max(0, C_{4,2} - d_4)\} \\ &= \min\{15 + 0,\ 6 + 1 + 0,\ 2 + 15 + 1 + 0\} = 7. \end{aligned}$$

Thus, $k^*(4,2) = 2$.

TABLE I
SOLUTIONS OF THE SIMPLE EXAMPLE

$\{T(j,s),\ k^*(j,s)\}$		s				
		1	2	3	4	5
j	1	(2,1)	{∞,-}	{∞,-}	{∞,-}	{∞,-}
	2	(6,2)	(6,1)	{∞,-}	{∞,-}	{∞,-}
	3	(15,3)	(6,1)	(6,1)	{∞,-}	{∞,-}
	4	{∞,-}	(7,2)	(9,1)	(14,1)	{∞,-}
	5	{∞,-}	(19,3)	(7,1)	(9,1)	(19,1)

Continuing on in this manner, we can have the results of Table I. Each cell of the table shows the optimal tardiness—$T(j,s)$, and its corresponding optimal last batch size—$k^*(j,s)$. From the last row of this table, we can find the optimal total tardiness $T^*(J) = T^*(5) = \min\{T(5,1),\ T(5,2),\ T(5,3),\ T(5,4),\ T(5,5)\} = \{\infty,\ 19,\ 7,\ 9,\ 19\} = 7$, where the optimal batch number (s^*) is 3. Then, we can now backward construct the optimal decisions:

1) $k^*(J,s^*) = k^*(5,3) = 1$; therefore, the last batch consists of only one mask—mask 5.
2) $k^*(5-1,3-1) = k^*(4,2) = 2$; therefore, the size of next-to-last batch is 2 and it contains mask 3 and 4.
3) $k^*(4-2,2-1) = k^*(2,1) = 2$; thus, the size of first batch is 2 and it includes mask 1 and 2.

We can clearly see from this simple example that the DP approach generates the best solution.

III. Dynamic Programming for Two Mask Sizes

The mask shop under study makes two mask sizes. Different mask sizes cannot be mixed in one batch; therefore, the above DP must be modified to solve the two-mask-size problem. In addition to the number of masks in the last batch, the mask type is another dimension of decision in each stage of the DP formulation for a two-mask-size problem.

We need additional notation to present the DP formulation for this problem.

The state variables to be used are

- s number of setups used;
- j_1 current type-1 mask number after sorting according to the EDD rule; $j_1 = 1,\ 2, \ldots, J_1$;
- j_2 current type-2 mask number after sorting according to the EDD rule; $j_2 = 1,\ 2, \ldots, J_2$.

The known parameters are

- $J_1,\ J_2$ total numbers of type-1 and type-2 masks to be scheduled, respectively;
- $p^1_{j_1},\ d^1_{j_1}$ processing time and the due date for the j_1th mask of type-1;
- $p^2_{j_2},\ d^2_{j_2}$ processing time and the due date for the j_2th mask of type-2;
- $C_{j_1,j_2,s}$ total processing time (completion time) for processing j_1 type-1 masks and j_2 type-2 masks with s setups.

The decision variables used are

- k_1 number of masks in the last batch, if it contains type-1 masks.
- k_2 number of masks in the last batch, if it contains type-2 masks.

The function notations are

- $\tilde{T}_1(j_1,j_2,s)$ minimum total tardiness, if processing the first j_1 type-1 masks and the first j_2 type-2 masks with s setups and the last batch containing type-1 masks;
- $\tilde{T}_2(j_1,j_2,s)$ minimum total tardiness, if processing the first j_1 type-1 masks and the first j_2 type-2 masks with s setups and the last batch containing type-2 masks;
- $T(j_1,j_2,s)$ minimum total tardiness for processing the first j_1 type-1 masks and the first j_2 type-2 masks with s setups.

The recursive function of $T(j_1,j_2,s)$ can be written as

$$T(j_1,j_2,s) = \min\{\tilde{T}_1(j_1,j_2,s), \tilde{T}_2(j_1,j_2,s)\}$$

where

$$\tilde{T}_1(j_1,j_2,s) = \min_{k_1=1,2,\ldots,b'_{j_1}} \left\{ T(j_1-k_1,j_2,s-1) + \sum_{l=j_1-k_1+1}^{j_1} \max\left(0, C_{j_1,j_2,s} - d^1_l\right) \right\}$$

$$\tilde{T}_2(j_1,j_2,s) = \min_{k_2=1,2,\ldots,b'_{j_2}} \left\{ T(j_1,j_2-k_2,s-1) + \sum_{l=j_2-k_2+1}^{j_2} \max\left(0, C_{j_1,j_2,s} - d^2_l\right) \right\}$$

and

$$C_{j_1,j_2,s} = s\cdot u + \sum_{i=1}^{j_1} p^1_i + \sum_{i=1}^{j_2} p^2_i$$
$$b'_{j_1} = \min\{b, j_1\}$$
$$b'_{j_2} = \min\{b, j_2\}.$$

As indicated by the above equation, the term $C_{j_1,j_2,s}$, which is the completion time for processing the first j_1 type-1 masks and the first j_2 type-2 masks in s setups, is independent of previous batch decisions. When finding the optimal solution of the first j_1 type-1 masks and the first j_2 type-2 masks, the minimum number and the maximum number of batches (setups) can be computed using

$$\underline{s} = \left\lceil \frac{j_1}{b} \right\rceil + \left\lceil \frac{j_2}{b} \right\rceil \quad \text{and} \quad \overline{s} = j_1 + j_2.$$

In addition, the boundary conditions are

$$T(j_1,j_2,s) = \infty, \quad \text{if } s < \underline{s} \quad \text{or } s > \overline{s}$$

and

$$T(j_1,j_2,s) = 0, \quad \text{if } j_1 = 0 \quad \text{and} \quad j_2 = 0.$$

The tardiness is infinitely large if the number of batches s is smaller than the minimum number of batches required or greater than the number of masks scheduled. Whereas the tardiness is zero if there are no masks requiring processing.

After defining the recursive relationship, the optimal total tardiness for the complete problem is obtained by computing

$$T^*(J_1, J_2) = \min_{s=\underline{s},\ldots,\bar{s}} \{T(J_1, J_2, s)\}$$

which determines the optimal number of batches (setups) needed to minimize total tardiness.

IV. Multiple Machine Problem

For a conventional identical-parallel-machine problem, the scheduling decision is normally divided into two parts: allocation and sequencing [9]. That is, we first allocate the jobs among machines; then, sequence the allocated jobs on each machine. This idea can be used for the scheduling of multiple E-beam writers. Several steps in the proposed heuristic approach are performed. First, all masks can be sorted according to due dates using the EDD dispatch policy. Second, the masks can be allocated to machines from the initial sequence using the *smallest-load machine rule*, as proposed by Baker and Merten [1] and Ho and Chang [9]. The smallest-load machine rule sequentially allocates jobs in the initial sequence to the machine with the smallest workload. This rule balances workloads among machines, and it has been shown to be effective in tardiness scheduling problems involving identical parallel machines [1], [9]. Third, the DP approach proposed in previous sections is used to make batch decision for each machine, to which the masks are allocated.

V. Computational Experiments

The solution program was written in the C programming language [11] and run on a computer workstation, directly linked to the mask shop E-beam writers as a part of the shop's CIM system. When considering the DP for the two-mask-size problem, instead of directly using the recursive-function feature in C language, we used a nested-loop-structure programming style with a two-dimensional array to store the partial tardiness solution. This makes the program very efficient. The solution time for a typical problem involving about 100 masks is less than 10 seconds on a SUN UNIX computer workstation. The solutions of this computer program were verified using another branch-and-bound program which, however, does not include a sharp bounding function. Therefore, it is much less efficient, and took more than 5 h to find the optimal solution of a much smaller 25-mask problem on the same computer workstation.

Examination of the computer program implementing the DP formulation shows the number of nested-loop levels is the same as the number of state variables in its formulation. Thus, the computation complexity is $O(n^2)$ for the one-mask-size problem and $O(n^3)$ for the two-mask-size problem. Due to the DP formulation and the effective nested-loop-structure programming techniques, the complexity of the algorithm is polynomial and the implemented program is very efficient, which allows constant rescheduling of the E-beam writers, whenever a new job arrives.

The performance of various approaches may be affected by different problem characteristics. Therefore, random problems were generated to do the comparisons. There were four changing factors in the experiments: various numbers of machines; various product ratios between 5- and 6-in masks; various demand rates; and various degrees of backlog.

Let

- m number of machines;
- $\bar{p}$ expected processing time per mask excluding batch setup time.
- $\bar{t}$ expected batch time with full batch size; i.e., $\bar{t} = u + b \cdot \bar{p}$
- $\bar{C}$ expected completion time of a machine; and

Based on the statistics from the mask shop, the average number of masks to be scheduled per machine is about 100; thus, the total number of masks (Q) to be scheduled is set as $100 \times m$. The maximum batch size (b) is 10 and the batch setup time (u) is 25 min.

The number of machines (factor 1) was varied from 1 to 5.

There were five different expected product ratio between 5- and 6-in masks (factor 2): $R_1 = (0.1\!:\!0.9)$, $R_2 = (0.3\!:\!0.7)$, $R_3 = (0.5\!:\!0.5)$, $R_4 = (0.7\!:\!0.3)$, and $R_5 = (0.9\!:\!0.1)$. For each mask in a random problem, a random value is generated to determine its mask size based on the product ratio of the random problem. The processing times of 5- and 6-in masks were randomly generated from a uniform distribution with range between 10 and 30 min and a uniform distribution with range between 20 and 150 min, respectively. Therefore, for example, the $\bar{p}$ is $0.1 \times \frac{10+30}{2} + 0.9 \times \frac{20+150}{2}$, when the product ratio is R_1. The due dates are also randomly generated from a uniform distribution; but, the range is determined by the factors of demand rate and the degree of backlog as outlined below.

The demand rates (factor 3) were divided into five levels by varying the average batch size when computing the expected completion time:

- Level D_1: average batch size $= 1$, thus $\bar{C} = 100 \times \bar{p} + 100 \times u$;
- Level D_2: average batch size $= \frac{b}{4}$, thus $\bar{C} = 100 \times \bar{p} + \frac{100}{b/4} \times u$;
- Level D_3: average batch size $= \frac{b}{2}$, thus $\bar{C} = 100 \times \bar{p} + \frac{100}{b/2} \times u$;
- Level D_4: average batch size $= \frac{3b}{4}$, thus $\bar{C} = 100 \times \bar{p} + \frac{100}{3b/4} \times u$;
- Level D_5: average batch size $= b$, thus $\bar{C} = 100 \times \bar{p} + \frac{100}{b} \times u$.

From the above formulas to compute $\bar{C}$, we can see that the higher demand level, the larger average batch size, and the smaller $\bar{C}$ value. The $\bar{C}$ value will be used as the range of the uniform distribution for generating the random due dates for the 100 masks. The smaller $\bar{C}$ value implies the higher demand rate, since the number of masks due within the $\bar{C}$ time interval is fixed at 100 in our experiment.

TABLE II
EXPERIMENT RESULTS OF ALL RANDOM PROBLEMS

N_{DP}	N_{DFB}	N_{FB}	$\frac{N_{DFB}}{N_{DP}}$	$\frac{N_{FB}}{N_{DP}}$
2.818	6.299	4.873	2.235	1.729

The backlog situations (factor 4) were divided into 5 levels by varying range of the uniform distributions for generating the due dates.

Level B_1: due dates were randomly generated from uniform$(\bar{t}, \bar{C} + \bar{t})$;
Level B_2: due dates were randomly generated from uniform$(\frac{1}{2}\bar{t}, \bar{C} + \frac{1}{2}\bar{t})$;
Level B_3: due dates were randomly generated from uniform$(0, \bar{C})$;
Level B_4: due dates were randomly generated from uniform$(-\frac{1}{2}\bar{t}, \bar{C} - \frac{1}{2}\bar{t})$;
Level B_5: due dates were randomly generated from uniform$(-\bar{t}, \bar{C} - \bar{t})$.

Note that the higher level has worse backlogging, since the due dates are earlier.

There were a total of $625\ (=5\times 5\times 5\times 5)$ parameter sets by varying the above four factors; each factor with five different values. For each parameter set, 30 random problems were generated. Thus, there were totally 18 750 random problems solved using the above three methods. A normalized metric of total tardiness was used for comparisons, defined as

$$N = \frac{T}{C},$$

where T is the total tardiness of a particular schedule and C is total processing time of all masks scheduled excluding setup times.

The experimental results are shown in Tables II–VI. In these tables, N_{DP} is the average normalized total tardiness of the schedules obtained by the Dynamic Programming approach; N_{DFB} is that obtained by the Dynamic Fixed Batch approach; and N_{FB} is that by the Full Batch approach. From the overall average results of all the random problems, Table II indicates that the DP approach is better than the FB approach, which in turn is better than the DFB approach. Using the DP approach, the tardiness is reduced on average by 55% from the DFB approach and the tardiness is reduced by 42% from the FB approach.

It is obvious and confirmed by the columns for N_{DP}, N_{DFB}, and N_{FB} in Table V and VI that the higher demand rate and the higher the degree of backlog, the larger the tardiness. The columns for N_{DP}, N_{DFB}, and N_{FB} in Table III shows that the normalized tardiness is smaller when there are more machines, no matter what methods we use. This observation could be explained by the "economies of scale".

By observing the columns of $\frac{N_{\mathrm{DFB}}}{N_{\mathrm{DP}}}$ and $\frac{N_{\mathrm{FB}}}{N_{\mathrm{DP}}}$ in Table IV, we can conclude that when there are more 6-in masks in the problem, the DP is more significant than the other two methods, since the values in these two columns are increased. From the columns for $\frac{N_{\mathrm{DFB}}}{N_{\mathrm{DP}}}$ and $\frac{N_{\mathrm{FB}}}{N_{\mathrm{DP}}}$ in Tables III, V, and VI, we can see that when the number of machine is larger, the demand rate is higher, and the backlog is higher, the performance differences between DP and the other two methods are less, since the values in these two columns are decreased. It is especially significant for the degree of backlog. Under the B_1 degree of backlog, the range of the uniform distribution used to generate the random due dates is shifted forward by the processing time of a full batch. This means the shop is operated with good order management; i.e., it only quotes delivery dates which are likely to be honored on time. To achieve this goal, the order management system has to

TABLE III
EXPERIMENT RESULTS BY VARYING THE NUMBER OF MACHINES NUMBER OF MACHINES

number of machines	N_{DP}	N_{DFB}	N_{FB}	$\frac{N_{DFB}}{N_{DP}}$	$\frac{N_{FB}}{N_{DP}}$
1	3.052	7.496	6.763	2.456	2.216
2	2.770	6.099	4.804	2.202	1.735
3	2.805	6.128	4.449	2.185	1.586
4	2.764	5.945	4.248	2.151	1.537
5	2.699	5.827	4.100	2.159	1.519

TABLE IV
EXPERIMENT RESULTS BY VARYING THE RATIO OF PRODUCT MIX RATIO (5-in : 6-in)

Ratio (5 inch : 6 inch)	N_{DP}	N_{DFB}	N_{FB}	$\frac{N_{DFB}}{N_{DP}}$	$\frac{N_{FB}}{N_{DP}}$
R_1 (0.1 : 0.9)	4.020	5.996	5.193	1.492	1.292
R_2 (0.3 : 0.7)	2.902	5.652	4.265	1.948	1.470
R_3 (0.5 : 0.5)	2.551	5.888	4.319	2.308	1.693
R_4 (0.7 : 0.3)	2.295	6.467	4.706	2.818	2.051
R_5 (0.9 : 0.1)	2.322	7.493	5.881	3.227	2.533

TABLE V
EXPERIMENT RESULTS BY VARYING THE DEMAND RATE LEVEL OF DEMAND RATE

level of demand rate	N_{DP}	N_{DFB}	N_{FB}	$\frac{N_{DFB}}{N_{DP}}$	$\frac{N_{FB}}{N_{DP}}$
D_1	0.884	4.665	1.825	5.278	2.064
D_2	2.131	6.485	3.847	3.043	1.805
D_3	3.229	6.373	5.520	1.974	1.709
D_4	3.769	6.824	6.347	1.811	1.684
D_5	4.076	7.148	6.826	1.754	1.675

TABLE VI
EXPERIMENT RESULTS BY VARYING THE DEGREE OF BACKLOG LEVEL OF BACKLOG

level of backlog	N_{DP}	N_{DFB}	N_{FB}	$\frac{N_{DFB}}{N_{DP}}$	$\frac{N_{FB}}{N_{DP}}$
B_1	0.090	2.078	0.765	23.084	8.496
B_2	0.533	3.087	1.877	5.795	3.523
B_3	1.918	5.163	3.945	2.692	2.057
B_4	4.273	8.516	6.984	1.993	1.634
B_5	7.275	12.650	10.794	1.739	1.484

consider the capacity consumption of current orders and the capacity limitation of the shop. Under this situation, the DP approach will dynamically adjust the batch size to achieve the minimum tardiness. We can see that, under the B_1 backlog level, the normalized total tardiness (N_{DP}) is only 0.09. In contrast, the normalized total tardiness of DFB approach (N_{DFB}) is 23 times of that of DP approach (N_{DP}). Therefore, when the shop is operated under good order management, the DP approach is much more significant in reducing the tardiness from levels characteristic of the other methods.

VI. Conclusion

Wafer fabrication is the most important portion for the entire semiconductor manufacturing. Mask shops make the key tools—masks—for the wafer fabrication. For most mask shops, the E-beam writer is the bottleneck owing to its high capital investment. This study presents a way of effectively scheduling the E-beam writers, which in fact plays a prominent role in semiconductor manufacturing.

This paper has presented a scheduling method for E-beam writers, which constitute a bottleneck for the mask shop under study. On-time delivery of masks is important to wafer fabrication factories, since they normally experience long production cycle times. If masks cannot be delivered on time, longer times will be required for wafer fabrication processes. Therefore, a due-date-related criterion—minimum total tardiness—was used in our approach to the scheduling problem. The factory management in our case study has established an Earliest-Due-Date dispatch rule. Thus, confirmed masks can be sorted according to their due dates. Then, based on the smallest-load machine rule, the masks can be allocated among parallel E-beam writers. Finally, the DP formulation proposed herein can be applied to each machine to calculate the optimal batching decisions. Considering the random problems experimented in this study, the proposed DP approach reduces the total tardiness by an average of 55% from the mask shop current scheduling method, and an average of 42% from the full-batch method. Furthermore, if the mask shop quotes realistic due dates, the DP approach will reduce the tardiness about 95% from the current method and about 88% from the full-batch method.

Acknowledgment

The author is grateful to the E-Beam Division of the Taiwan Semiconductor Manufacturing Company for providing the experiment data and is also indebted to Professor R. C. Leachman from University of California at Berkeley for comments on this manuscript.

References

[1] K. R. Baker and A. G. Merten, "Scheduling with parallel processors and linear delay costs," *Naval Res. Logist. Quart.*, vol. 20, pp. 793–804, 1973.
[2] V. Chandru, C.-Y. Lee, and R. Uzsoy, "Minimizing total completion time on a batch processing machine with job families," *Oper. Res. Lett.*, vol. 13, pp. 61–65, 1993.
[3] ——, "Minimizing total completion time on batch processing machines," *Int. J. Prod. Res.*, vol. 31, no. 9, pp. 2097–2121, 1993.
[4] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*. Orlando, FL: Academic, 1977, p. 17.
[5] J. Du and J. Y.-T. Leung, "Minimizing total tardiness on one machine is NP-hard," *Math. Oper. Res.*, vol. 15, no. 3, pp. 483–495, Aug. 1990.
[6] J. W. Fowler, D. T. Phillips, and G. L. Hogg, "Real-time control of multiproduct bulk service semiconductor manufacturing processes," *IEEE Trans. Semiconduct. Manufact.*, vol. 5, pp. 158–163, 1992.
[7] C. R. Glassey and W. Weng, "Dynamic batching heuristics for simultaneous processing," *IEEE Trans. Semiconduct. Manufact.*, vol. 4, pp. 77–82, 1991.
[8] F. S. Hillier and G. J. Liberman, *Introduction to Operations Research*. New York: McGraw Hill, p. 431, 1995.
[9] J. C. Ho and Y.-L. Chang, "Heuristics for minimizing mean tardiness for m parallel machines," *Naval Res. Logist.*, vol. 38, pp. 367–381, 1991.
[10] Y. Ikura and M. Gimple, "Efficient scheduling algorithms for a single batch processing machine," *Oper. Res. Lett.*, vol. 5, no. 2, pp. 61–65, 1986.
[11] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[12] C. Koulamas, "The total tardiness problem: Review and extensions," *Oper. Res.*, vol. 62, no. 6, pp. 1025–1041, Nov.–Dec. 1994.
[13] C.-Y. Lee, R. Uzsoy, and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Oper. Res.*, vol. 40, pp. 764–775. 1992.
[14] R. Uzsoy, "Scheduling a single batch processing machine with non-identical job sizes," *Int. J. Prod. Res.*, vol. 32, no. 7, pp. 1615–1635, 1994.
[15] H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot size model," *Manage. Sci.*, vol. 5, pp. 89–96, 1958.
[16] W. W. Weng and R. C. Leachman, "An improved methodology for real-time production decisions at batch process work stations," *IEEE Trans. Semiconduct. Manufact.*, vol. 6, pp. 219–225, Aug. 1993.

Yi-Feng Hung was born in Tainan, Taiwan, R.O.C., in 1961. He received the B.S. degree in industrial engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1984, and the M.S. degree in industrial engineering and the Ph.D. in industrial engineering and operations research from the University of California at Berkeley, in 1989 and 1991, respectively.

He is an Associate Professor of Industrial Engineering at the National Tsing Hua University. His research interests include production management, simulation application, and factory database design for semiconductor manufacturing. He currently works closely with the semiconductor companies in Science-based Industrial Park, Hsinchu, Taiwan on issues of manufacturing management.

Dr. Hung received the Outstanding Industrial Collaboration Award from the Ministry of Education in 1997.